Modified Kernel-based Nonlinear Feature Extraction

J. Ma, S. Perkins, J. Theiler
NIS-2,
Los Alamos National Lab,
Los Alamos, NM, USA

S. Ahalt
Department of Electrical Engineering
Ohio State University
Columbus, OH, USA

Abstract:
Feature Extraction (FE) techniques are widely used in many applications to pre-process data in order to reduce the complexity of subsequent processes. A group of Kernel-based nonlinear FE (KFE) algorithms has attracted much attention due to their high performance. However, a serious limitation that is inherent in these algorithms -- the maximal number of features extracted by them is limited by the number of classes involved -- dramatically degrades their flexibility. Here we propose a modified version of those KFE algorithms (MKFE). This algorithm is developed from a special form of scatter-matrix, whose rank is not determined by the number of classes involved, and thus breaks the inherent limitation in those KFE algorithms. Experimental results suggest that MKFE algorithm is especially useful when the training set can only sparsely represent the distribution of the underlying problem.

Keywords: *kernel-based feature extraction (KFE), kernel trick, modified kernel-based feature extraction (MKFE), nonlinear feature extraction, support vector machines (SVMs).*

1. Introduction

Feature Extraction (FE) plays a pivotal role in many applications such as pattern recognition and data mining. Among a large number of FE algorithms developed over the past three decades, a group of kernel-based nonlinear FE algorithms stand out recently due to their high performance [1-5]. Two typical examples of this class of algorithms are the Kernel Fisher Discriminant (KFD) algorithm proposed by Mika, et. al. [1], and the Kernel-based nonlinear FE (KFE) algorithm proposed by Ma [4]. Both algorithms employed a technique referred to as the "kernel trick" to introduce non-linearity into the well-established linear algorithm. KFD is a nonlinear extension of the Fisher's criterion, and was mainly developed for two-classes problem, while KFE is a nonlinear extension of the linear discriminant analysis (LDA), and is applicable to multiple-class problem.

However, both algorithms can only extract at most *L-1* features, where *L* is number of classes involved. In this paper we propose a modified algorithm, which breaks the above limitation. Experimental results suggest that this Modified Kernel-based nonlinear Feature Extraction (MKFE) algorithm is more useful for smaller training set.

Because the MKFE algorithm is a direct development of the KFE algorithm, in order to facilitate the reader's understanding, we will briefly describe the KFE algorithm in Section 2. The MKFE algorithm is derived and presented in Section 3. The performance of the MKFE algorithm is demonstrated by a real-world experiment in Section 4.

2. Kernel-based Nonlinear Feature Extraction

A linear FE problem can be defined as follows: If we want to construct the optimal *m*-feature new patterns from original *n*-feature patterns given a criterion *J*, we need to find an *n*-by-*m* matrix **E**, which generates an optimal *m*-feature sample, **R**, from the original *n*-feature sample, **X**. That is, $\mathbf{R} = \mathbf{E}^T\mathbf{X}$ [7].

Thus, selecting a criterion *J* to quantitatively measure optimal class-separability is generally a prerequisite for developing FE algorithm. Although Bayes classification error, P_e, is a natural criterion, its difficulty of estimation [6] makes its direct application impractical. Due to its simplicity, intuitive appeal, relatively good performance, and robustness, an alternative, scatter-matrix based linear criteria is widely adopted [7]. This criterion can be represented as:

$$J_1 = tr(\mathbf{S}_w^{-1}\mathbf{S}_b)\,, \tag{1}$$

where $tr\{\mathbf{A}\}$ denotes the trace operation of a matrix **A**. $\mathbf{S}_w$ is the within-matrix, and indicates the spread of samples, $\mathbf{X}_k^{(i)}$, around the individual class mean, $\mathbf{M}_i$.

$$\mathbf{S_w} = \sum_{i=1}^{L}\frac{N_i}{N}\sum_{k=1}^{N_i}(\mathbf{X}_k^{(i)} - \mathbf{M}_i)(\mathbf{X}_k^{(i)} - \mathbf{M}_i)^T \tag{2a}$$

$\mathbf{S}_b$ is the between-class matrix, and indicates the spread of mean of each class, $\mathbf{M}_i$, around the mean of all the classes, $\mathbf{M}_0$.

$$\mathbf{S_b} = \sum_{i=1}^{L}\frac{N_i}{N}(\mathbf{M_i} - \mathbf{M_0})(\mathbf{M_i} - \mathbf{M_0})^T\,. \tag{2b}$$

The solution matrix **E** to the linear FE problem given criterion J_1 is therefore [7]:

$$\mathbf{E} = [e_1, e_2, \cdots e_m]_{n\times m}\,, \tag{3}$$

where $\mathbf{e}_i$, *i=1…m*, are eigenvectors of matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ corresponding to the *m* largest eigenvalues.

This linear FE algorithm has three critical drawbacks: (a) It will fail when the sample mean, M_i, of any class *i*, is the same as the sample mean of all samples, M_0; (b) This FE algorithm can only exploit linear class separability; (c) It can only extract *L-1* features for a *L*-class problem. The first two drawbacks motivate us to extend this linear FE algorithm to the nonlinear domain, while the third one is handled in the next section. The technique we employed here is named "*kernel trick*"[10]. The basic idea of the kernel trick is to simply replace the dot product in a Euclidean space with a nonlinear kernel function. That is:

$$<\mathbf{X},\mathbf{Y}>= \mathbf{X}^T\mathbf{Y} \to K(\mathbf{X},\mathbf{Y})\,, \tag{4}$$

where the kernel function, *K(*X,Y*)*, maps a subspace in ($R^n \times R^n$) to a subspace in *R*. This transformation can be interpreted as follows[8]:

(1) Map a sample *X* from a subspace of R^n, or *input space, I*, to another Euclidean space, or *feature space*, *H*, using a functional vector, $\mathbf{\Phi}$:

$$\mathbf{\Phi} : I \to H\,.$$

(2) The kernel function, *K*(*X,Y*), can be defined as a dot-product in space *H*. That is,

$$K(\mathbf{X},\mathbf{Y}) = \mathbf{\Phi}(\mathbf{X})^T\,\mathbf{\Phi}(\mathbf{Y}) \tag{5}$$

Therefore, a kernel function implicitly introduces both a Euclidean space *H*, and a map $\mathbf{\Phi}$.

The general formulation of a nonlinear FE problem can be defined as follows:

$$\mathbf{R} = \mathbf{F}(\mathbf{X}) = \begin{bmatrix} F_1(\mathbf{X}) \\ F_2(\mathbf{X}) \\ \vdots \\ F_m(\mathbf{X}) \end{bmatrix}, \tag{6}$$

where *F* is a nonlinear functional vector mapping a n-feature sample, *X*, to a m-feature extracted sample, *R*;

According to the theory of reproducing kernel, we can represent the nonlinear FE function, $F_i(X)$, *i=1…m*, in Equation (6) as a linear combination of a group of *kernel* *functions*, $K(X_j,X)$, $j=1 \ldots N$, the general form of the nonlinear FE (6) can be expressed as:

$$\mathbf{R} = \mathbf{F}(\mathbf{X}) = \begin{bmatrix} \sum_{i=1}^{N} \alpha_{i,1} K(\mathbf{X}_i, \mathbf{X}) \\ \sum_{i=1}^{N} \alpha_{i,2} K(\mathbf{X}_i, \mathbf{X}) \\ \vdots \\ \sum_{i=1}^{N} \alpha_{i,m} K(\mathbf{X}_i, \mathbf{X}) \end{bmatrix} \qquad (7)$$

$$= \mathbf{A}^T \mathbf{K}(\bullet, \mathbf{X})$$

where

$$\mathbf{A} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N,1} & \alpha_{N,1} & \cdots & \alpha_{N,m} \end{bmatrix}$$

$$= [\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_m]$$

$\alpha_{i,j}$ are a set of coefficients;

and

$$\mathbf{K}(\bullet, \mathbf{X}) = \begin{bmatrix} K(\mathbf{X}_1, \mathbf{X}) \\ K(\mathbf{X}_2, \mathbf{X}) \\ \vdots \\ K(\mathbf{X}_N, \mathbf{X}) \end{bmatrix},$$

where X_i ($i=1 \ldots N$) are a set of representative samples, or the training set.

Thus developing a nonlinear FE algorithm is equivalent to finding a matrix, A, which maximizes the class separability of the extracted samples, R.

Substituting Equation (5) into (7) we obtain:

$$\mathbf{R} = \begin{bmatrix} \sum_{i=1}^{N} \alpha_{i,1} \mathbf{\Phi}(\mathbf{X}_i)^T \\ \sum_{i=1}^{N} \alpha_{i,2} \mathbf{\Phi}(\mathbf{X}_i)^T \\ \vdots \\ \sum_{i=1}^{N} \alpha_{i,m} \mathbf{\Phi}(\mathbf{X}_i)^T \end{bmatrix} \mathbf{\Phi}(\mathbf{X}) . \qquad (8)$$

$$= \mathbf{E}_F{}^T \mathbf{\Phi}(\mathbf{X})$$

The nonlinear FE defined in Equation (6) is thus reformulated to be a linear FE in the *feature space, H*, which enables the established linear FE solution (3) readily applicable.

Now, we define *kernel within-matrix,* G_w*,* for a set of samples, X_i, $i = 1 \ldots N$, as:

$$\mathbf{G}_w = \mathbf{\Theta}^T_{\Phi(X)} \mathbf{S}_{w\Phi(X)} \mathbf{\Theta}_{\Phi(X)} \qquad (9)$$

and define *kernel between-class matrix,* G_b*,* for a set of patterns X_i, $i = 1 \ldots N$, as:

$$\mathbf{G}_b = \mathbf{\Theta}^T_{\Phi(X)} \mathbf{S}_{b\Phi(X)} \mathbf{\Theta}_{\Phi(X)} \qquad (10)$$

where $\mathbf{\Theta}_{\Phi(X)} = \left[\mathbf{\Theta}^{(1)}_{\Phi(X)} \quad \cdots \quad \mathbf{\Theta}^{(L)}_{\Phi(X)}\right]$ and $\mathbf{\Theta}^{(i)}_{\Phi(X)} = \left[\mathbf{\Phi}(\mathbf{X}^{(i)}_1) \quad \cdots \quad \mathbf{\Phi}(\mathbf{X}^{(i)}_{N_i})\right]$.

It is easy to show that the rank of matrix G_w is not bigger than N-L, while the rank of matrix G_b is not bigger than L-1, where N is number of samples in the training set, and L is number of classes. Therefore, in order to make *kernel within-matrix,* G_w, invertible, a *conditioned kernel within-matrix*, $\overline{G}_W$, is introduced as:

$$\overline{\mathbf{G}}_w = \mathbf{G}_w + \tau \mathbf{I} \qquad (11)$$

where $\tau > 0$ and is called the *conditioning coefficient*, and I is a identity matrix.

The KFE algorithm can thus be obtained by applying (5), (7), (8), (9), (10), and (11) to (3), as well as some algebra manipulation. The algorithm can be finally described as:

The matrix A in (7) that maximizes criterion J_1 *in (1) can be formed by m eigenvectors corresponding to the m largest eigenvalues of matrix* $\overline{G}_w^{-1} G_b$*. That is:*

$$\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_m] \qquad (12)$$

where $(\overline{\mathbf{G}}_w^{-1} \mathbf{G}_b) \mathbf{a}_i = \lambda_i \mathbf{a}_i$*,* $i = 1 \ldots m$*, and* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$.

3. Modified Kernel-based Feature Extraction

Because $Rank(G_b) \leq L$-1, the KFE algorithm mentioned previously can only extract a maximum of L-1 meaningful features even if m is larger than L-1. This limitation motivates us to modify the KFE algorithm to allow it to extract up to N

features, where *N* is the number of samples in the training set. This is achieved by modifying the formulation of the *kernel between-class matrix* G_b to increase its rank. In order to simplify our following derivation, we assume the underlying problem is a two-class problem, while the fundamental idea can be readily extended to multi-class problems.

Fortunately, a result from statistical discriminant analysis, *nonparametric between-class matrix,* $\mathbf{S}_{\mathrm{nb}}$, provides us with an alternative to the *between-class matrix* $\mathbf{S}_{\mathrm{b}}$ [7]. The estimate of S_{nb} can therefore be expressed as:

$$\mathbf{S}_{n\mathbf{b},X} = \frac{1}{N}$$

$$\Big(\sum_{i=1}^{N_1} w_{X_i^{(1)},k}[\mathbf{X}_i^{(1)} - \mathbf{M}_k^{(2)}(\mathbf{X}_i^{(1)})][\mathbf{X}_i^{(1)} - \mathbf{M}_k^{(2)}(\mathbf{X}_i^{(1)})]^T +$$

$$\sum_{i=1}^{N_2} w_{X_i^{(2)},k}[\mathbf{X}_i^{(2)} - \mathbf{M}_k^{(1)}(\mathbf{X}_i^{(2)})][\mathbf{X}_i^{(2)} - \mathbf{M}_k^{(1)}(\mathbf{X}_i^{(2)})]^T\Big) \tag{13}$$

where k is the number of neighbor patterns used, and $\mathbf{M}_k^{(1)}(\mathbf{X}_i^{(2)})$ is the estimated mean of the k *nearest neighbor* (*NN*) to sample $\mathbf{X}_i^{(2)}$, or $\mathbf{M}_k^{(1)}(\mathbf{X}_i^{(2)}) = \frac{1}{k}\sum_{a=1}^{k}\mathbf{X}_{aNN}^{(1)}$, where $\mathbf{X}_{aNN}^{(1)}$ is the ath *nearest neighbor* (*NN*) in class *1* to the sample $\mathbf{X}_i^{(2)}$ in class *2*. $\mathbf{M}_k^{(2)}(\mathbf{X}_i^{(1)})$ can be interpreted accordingly.

$$w_{X,k} = \frac{1}{\beta_{X,k}+1}, \text{ where}$$

$$\beta_{X,k} = \frac{\max(\| \mathbf{X} - \mathbf{X}_{kNN}^{(1)} \|^2, \| \mathbf{X} - \mathbf{X}_{kNN}^{(2)} \|^2)}{\min(\| \mathbf{X} - \mathbf{X}_{kNN}^{(1)} \|^2, \| \mathbf{X} - \mathbf{X}_{kNN}^{(2)} \|^2)}.$$

$w_{X,k}$ is a coefficient that more heavily weights the patterns falling in the boundary area between two classes. From (13), we know that the rank of S_{nb} is determined by the internal structure of the samples in the training set, and is not limited by the number of classes involved.

We therefore define the *kernel nonparametric between-class matrix* G_{nb}, for samples X_i, $i = 1...N$, is definied as:

$$\mathbf{G}_{nb} = \mathbf{\Theta}_{\Phi(X)}^T \mathbf{S}_{nb,\Phi(X)} \mathbf{\Theta}_{\Phi(X)} \tag{14}$$

where $\mathbf{\Theta}_{\Phi(X)} = \left[\mathbf{\Theta}_{\Phi(X)}^{(1)} \;\cdots\; \mathbf{\Theta}_{\Phi(X)}^{(L)}\right]$ and $\mathbf{\Theta}_{\Phi(X)}^{(i)} = \left[\mathbf{\Phi}(\mathbf{X}_1^{(i)}) \;\cdots\; \mathbf{\Phi}(\mathbf{X}_{N_i}^{(i)})\right]$.

By substituting (5) and (13) into (14), we obtain:

$$\mathbf{G}_{nb} = \frac{1}{N}\mathbf{DPD}^T, \tag{15}$$

where

$$P = \begin{bmatrix} \rho_{1,k}^{(1)} & & & & & \\ & \ddots & & & & \\ & & \rho_{N_1,k}^{(1)} & & & \\ & & & \rho_{1,k}^{(2)} & & \\ & & & & \ddots & \\ & & & & & \rho_{N_2,k}^{(2)} \end{bmatrix},$$

where $\rho_{l,k}^{(i)} = \dfrac{1}{\beta_{X_l^{(i)},k}+1}$,

and

$$\beta_{X_l^{(i)},k} =$$

$$\frac{\max(r_K(\mathbf{X}_l^{(i)} - \mathbf{X}_{kNN}^{(1)}), r_K(\mathbf{X}_l^{(i)} - \mathbf{X}_{kNN}^{(2)}))}{\min(r_K(\mathbf{X}_l^{(i)} - \mathbf{X}_{kNN}^{(1)}), r_K(\mathbf{X}_l^{(i)} - \mathbf{X}_{kNN}^{(2)}))},$$

and

$$r_K(\mathbf{X}_l^{(i)} - \mathbf{X}_{kNN}^{(1)}) =$$

$$K(\mathbf{X}_l^{(i)}, \mathbf{X}_l^{(i)}) + K(\mathbf{X}_{kNN}^{(1)}, \mathbf{X}_{kNN}^{(1)})$$

$$- 2K(\mathbf{X}_l^{(i)}, \mathbf{X}_{kNN}^{(1)})$$

$\mathbf{D} = \left[\mathbf{d}_1^{(1)} \;\cdots\; \mathbf{d}_{N_1}^{(1)} \;\; \mathbf{d}_1^{(2)} \;\cdots\; \mathbf{d}_{N_2}^{(2)}\right]$, where

$$d_l^{(i)} = \mathbf{K}(\bullet, \mathbf{X}_l^{(i)}) -$$

$$\frac{1}{k}[\mathbf{K}(\bullet, \mathbf{X}_{1NN}^{\sim(i)}) \;\cdots\; \mathbf{K}(\bullet, \mathbf{X}_{kNN}^{\sim(i)})]\mathbf{1}_{k\times 1}.$$

Note that $\mathbf{1}_{k\times 1}$ is a column vector, whose elements are all one, *~(i)* denotes the class different from class *i*, and $K(\bullet, X_i^{(i)})$ is defined in (7).

Therefore, the MKFE can be obtained simply by replacing the G_{b} in the KFE algorithm with G_{nb}, as defined in (15). The MKFE can be thus described as:

The matrix A in (7) that maximizes criterion J_1 in (1) can be formed by m eigenvectors corresponding to the m largest eigenvalues of matrix $\overline{\mathbf{G}}_w^{-1}\mathbf{G}_{nb}$. That is:

$$\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_m] \tag{16}$$

where $(\overline{\mathbf{G}}_w^{-1}\mathbf{G}_{nb})\mathbf{a}_i = \lambda_i \mathbf{a}_i$, i = 1...m, and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$.

4. Experiment

In order to test the efficacy of our MKFE algorithm, we applied it to a set of two-class target signatures obtained using High Range Resolution (HRR) radar. The kernel function employed in the MKFE algorithm is an RBF kernel function with $\gamma = 1$, and the number of *NN*s, or *k* in (15), is set to 3 empirically. RBF-based SVM classifiers [9] are employed to classify the new samples constructed by the MKFE algorithms from the original HRR signatures. We took 15% and 20% percentage of HRR signatures from the whole data set to use as the training set in two different experiments respectively, and used the corresponding remaining samples as the testing set. We repeated the experiment 100 times over 100 different random realizations of the training set and testing set to reduce the statistical variance, and the mean classification rates are plotted out in Figure 1.

As we mentioned previously, because this is a two-class problem, both KFD and KFE algorithms can only extract a single feature for each new samples. In contrast, the number of extracted features in each new samples contracted by MKFE is only limited by the number of original HRR signatures in the training set. In our experiment, we set MKFE to extract from 1 to 25 features for each new sample. We know that, when MKFE only extracts 1-feature, it is equivalent to the KFE algorithm. Thus, we can consider the results in Figure 1 when m=1 as the performance of the original KFE algorithm on this two-class problem. In this way, Figure 1 demonstrates that the MKFE enhances the performance of KFE by increasing the number of extracted features for each new sample.

From Figure 1, we can also observe that the MKFE outperforms KFE by about 2% when 20% of the whole HRR signature set is used as the training set, while it outperforms KFE by 4% when only 15% of the whole HRR signature set is used as the training set. When m=25, we can see that the MKFE algorithm makes the classification rate obtained when only 15% signatures are used as training set pretty close to the classification rate obtained when 20% signatures are used as training set. This observation suggests that our proposed MKFE algorithm is especially useful when the training information is limited, or the size of training set is small.

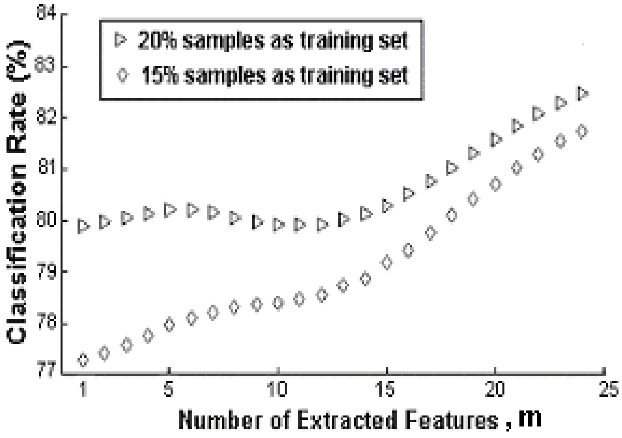

Figure 1. Classification rates vs. Number of Extracted Features, m

5. Conclusions

In this paper we have proposed a modified version of earlier KFE algorithms.

This algorithm is based on a special form of scatter-matrix, whose rank is not determined by the number of classes involved, and thus breaks the inherent limit in those KFE algorithms. Experiment results suggest that our proposed MKFE algorithm is especially useful when the training set is small.

In order to make this algorithm more practical, there are several open lines of research, including 1) how to refine this algorithm to make it less computationally demanding, 2) how to theoretically guide users in selecting a set of optimal algorithm parameters for a particular problem.

6. Acknowledgements

This work was first supported by the Mission Research Cooperation under subcontract SC-1406-98-0001. Additional support has been provided by a NASA project NRA-00-01-AISR-088, and a LANL LDRD project (Real World Machine Learning).

References:

[1] S. Mika, G. Ratsch, and J. Weston, "Fisher Discriminant Analysis With Kernels," *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pp. 41-48, Madison, Wisconsin, August 1999.

[2] G. Baudat, and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach," *Neural Computation*, vol. 12, no. 10, pp. 2385-2404, 2000.

[3] V. Roth, and V. Steinhage, "Nonlinear Discriminant Analysis Using Kernel Functions," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K. -R. Muller, Eds 1999, vol. 12, pp. 568-574, MIT Press.

[4] J. Ma, "Feature Study of High Range Resolution Based Automatic Target Recognition: Analysis and Extraction," *Ph. D. Dissertation*, the Ohio State University, June 2001.

[5] A. Ruiz, and P. E. Lopez-de-Teruel, "Nonlinear Kernel-Based Statistical Pattern Analysis," *IEEE Transactions on Neural Networks*, vol. 12, no. 1, pp. 16-32, January 2001.

[6] P. R. Krishnaiah, and L. N. Kanal, *Handbook of Statistics, vol. 2*, New York: North-Holland Publishing Company, 1982.

[7] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd Edition, Boston: Academic Press, c1990.

[8] B. Schölkopf, C. Burges, and A. Smola, *Advances in Kernel Methods: Support Vector Learning*, Cambridge: MIT Press, 1999

[9] J. Ma, H. Li, and S. Ahalt, "Using Support Vector Machines as HRR Signature Classifiers," *Proceedings of SPIE International Symposium on Automatic Target Recognition XI*, Orlando, FL, April 2001.

[10] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2(2), 1998.